

Jonathan Fontana | IT Manager DevOps & performance

## How PMU integrates the performance test into its DevOps platform using NeoLoad



### Context

PMU is the European leader in online betting and ranked in the top three worldwide. Regardless of how the bets are placed – online or via the traditional window, the company's digital technology, comprised of a centralized network, and applications developed in-house, is critical to its success as a whole.

The performance of PMU's applications is not only vital for around the clock uninterrupted activity but for guaranteeing the performance and reliability required during bet placement – typically done within a short window at high volumes throughout the day.

In order to meet customer expectations and continuously offer new uses, PMU has multiplied its release frequency threefold, changing from a quarterly rate to a monthly rate. In addition, the teams have succeeded in integrating the performance test into their continuous integration Agile process and in improving software performance management within the DevOps teams. A success that PMU attributes to the orchestration of tools, processes, and teams working in unison.

### Agile Development and Performance Test

Traditionally, PMU developed applications based on V-models with teams working in silos. On one side, the developers (Dev) and on the other, the operations (Ops) in charge of pre-production performance tests.

The lack of collaboration between Dev and Ops created friction in the development process. For example, performance testers discovered the applications to be tested when they received them from the Dev, without any context. This situation where the tester does not know what he needs to measure, or what he needs to test, generates a significant amount of toing and froing between Dev and testers, making the application validation phase considerably longer. Result: some applications were only released after one year of development.

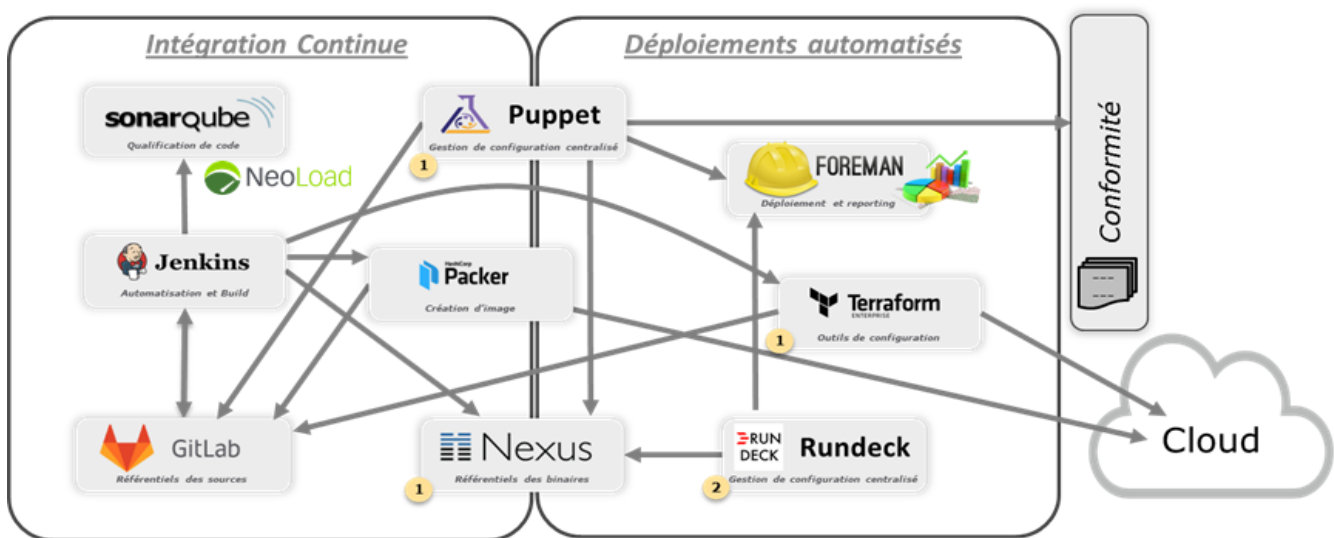
### The DevOps Platform Breaks Down the Wall

In order to meet the release cycle speed requirement, PMU set up a DevOps organization supported by a suitable tools platform. NeoLoad carries out the performance and load test function, with a maximum degree of test task automation in order to reduce the release cycles.

The term, DevOps platform, is important because it clearly shows willingness to support a collaborative process where each employee can contribute and share information with the rest of the team. Thanks to this platform, earlier silos have disappeared. When a developer produces a code, he pushes it into GIT, a Jenkins job starts a Sonar analysis, and if approved, a package will be created and delivered in Nexus. At the same time, this package is automatically installed on the various testing environments. The Ops teams see the code arrive in the tools that they are using, in a collaborative way.

This collaborative work, made possible in particular thanks to the NeoLoad plugin for Jenkins, means that all testing scenarios can be found in Jenkins. There is an option for launching tests individually or

joining forces with another user to do this, as Jenkins is connected to the LDAP (Lightweight Directory Access Protocol) for managing rights. History is available so that you can find out if the tests have been launched by a human being or automatically. PMU also has the option of making several scenarios usable by various employees. This works like a performance test service catalogue. In this way, the organization can track the development of the performance, check which builds have failed the tests, why and know how to remedy them. Lastly, the reporting can be sent to the relevant employees via a Jenkins email plugin.



With its DevOps platform, PMU is now able to carry out testing during acceptance phases.

Scenarios are modifiable but created upstream, which avoids reaching pre-production and having to backtrack. Acceptance and Dev development cycles are therefore much faster and more iterative.

### Automation: Time-To-Market Catalyst

One of the main advantages of automated tests is placing new functionalities and improvements on the market faster.

Jenkins jobs will update the tests in NeoLoad automatically. Platform maintenance is also done automatically using Rundeck, which enables tests to be run at night and over the weekend. This time saving due to night and weekend testing has multiplied the ability to produce results twofold.

The application chain changes on a daily basis and is therefore tested every day from pre-production, or even, upon acceptance if the analysis needs to be enhanced.

Prior to automation, if a delivery needed a specific test, the test would need to be done the day before the delivery and then would take one day to be qualified after delivery. Thus, pre-production was monopolized for two days. Night testing means that a reference can be obtained every night. Therefore, a test before delivery, along with the scheduling of a test afterwards, is no longer necessary because it is already scheduled to be carried out that very same evening. Consequently, business tests can be done during pre-production, during the day, without conflicting with the performance tests.



By implementing a DevOps process, PMU has therefore been able to save a considerable amount of time on performance tests:

- **Anticipating user paths** avoids iterations during pre-production. Tests are already carried out in upstream environments. Therefore, it is possible to know if everything is working correctly or if problems are arising.
- **Sharing Virtual Users** with all of the platform's users allows the team to continue working at the same time whilst continuing the real tests, beyond the implementation of ongoing tests.
- **Jenkins automation** means that weekends can be used to carry out non-regression performance tests since no one is working on the pre-production at that moment. The entire software part has been automated, as well as the performance part: every weekend, identical race schedules are injected and checked from weekend to weekend to make sure that there is no performance and functional regression regarding that scope.
- **Using weekends for non-regression software and performance tests** saves a considerable amount of time, and, frees up a huge amount of availability during the week for other tests when everyone is there to carry out analyses at the same time.

All night tests are currently automated thus saving 30% of the test slots. Managing several projects at the same time on several environments has therefore increased the teams' ability to test their reactivity during the performance test. The number of tests has tripled: currently, PMU can launch more than ten performance tests over one day where before it only did a maximum of three. This jives with the DevOps automation and agility process that is PMU is continually developing. A process that is also leading to a transformation towards the cloud.

As a result, in an increasingly competitive world, PMU is ensuring its leading position by offering a monthly release instead of the quarterly one that was needed in the past.